

日本国特許庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日 2001年 5月30日
Date of Application:

出願番号 特願2001-161754
Application Number:

パリ条約による外国への出願
用いる優先権の主張の基礎
となる出願の国コードと出願
号
The country code and number
of your priority application,
to be used for filing abroad
under the Paris Convention, is

J P 2001-161754

出願人 高振宇
Applicant(s):

CERTIFIED COPY OF
PRIORITY DOCUMENT

2005年11月30日

特許庁長官
Commissioner,
Japan Patent Office

中嶋



【書類名】 特許願

【整理番号】 K001-01

【提出日】 平成13年 5月30日

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 15/21

【発明者】

【住所又は居所】 埼玉県川口市金山町1番4-205号シャトープリンス

【氏名】 高 振宇

【特許出願人】

【識別番号】 593221598

【氏名又は名称】 高 振宇

【国籍】 中華人民共和国

【代理人】

【識別番号】 100093517

【弁理士】

【氏名又は名称】 豊田 正雄

【先の出願に基づく優先権主張】

【出願番号】 特願2000-299305

【出願日】 平成12年 9月29日

【手数料の表示】

【予納台帳番号】 030524

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 ホームページ改竄防止システム

【特許請求の範囲】

【請求項 1】

- (1) ウェブファイルを暗号化処理した暗号化ウェブファイルを格納する公開ウェブサーバーコンピュータ、
- (2) 前記公開ウェブサーバーコンピュータとファイアウォール等の不正アクセスを排除する手段を介して接続し、前記ウェブファイルを格納する原本ウェブサーバーコンピュータ、
- (3) 公開ウェブサーバーコンピュータにおけるウェブサーバーが、ユーザーからアクセス要求を受けたときに前記暗号化されたウェブファイルを改竄チェックを行い、改竄されていないと判断した場合、前記暗号化されたウェブファイルを復号化してユーザーに送信する手段、
- (4) 前記公開ウェブサーバーが前記暗号化されたウェブファイルの改竄を検出したとき、前記原本ウェブサーバーコンピュータで、記憶装置に格納されている対応するウェブファイルを暗号化処理して作成した暗号化ウェブファイルにより、前記公開ウェブサーバー・コンピュータの記憶装置へ送信して改竄されたウェブファイルを更新・復旧処理する手段、
を含むことを特徴とするホームページ改竄防止システム。

【請求項 2】

- (1) ウェブファイルに全般の認証を行う認証子を含むデータをヘッダーとして付加した改竄防止ヘッダー付きウェブファイルを格納する改竄防止機能付き公開ウェブサーバーコンピュータ、
- (2) 前記改竄防止機能付き公開ウェブサーバーコンピュータと不正アクセスを排除するファイアウォール等の不正アクセスを排除する手段を介して接続し、前記ウェブファイルを格納する原本ウェブサーバーコンピュータ、
- (3) 前記公開ウェブサーバーコンピュータにおけるウェブサーバーが、ユーザーからアクセス要求を受けたとき、前記改竄防止ヘッダー付きウェブファイルから該ヘッダーを分離し、該ヘッダー情報中の認証子で分離されたウェブファイル

に対して改竄チェックをするリアルタイムチェック手段、

(4) 前記公開ウェブサーバーが、前記リアルタイムチェックにより前記ウェブファイルが改竄されていないと判断した場合、前記ヘッダーを除去したウェブファイルをユーザーに送信する手段、

(5) 前記改竄防止ヘッダー付きウェブファイルの改竄を検出したとき、前記原本ウェブサーバーコンピュータで、記憶装置に格納されている対応するウェブファイルに改竄防止ヘッダーを附加して作成した改竄防止ヘッダー付きウェブファイルにより、前記改竄防止機能付き公開ウェブサーバーコンピュータの記憶装置へ送信して改竄されたウェブファイルを更新・復旧処理する手段、
を含むことを特徴とするホームページ改竄防止システム。

【請求項 3】

(1) ウェブファイルを暗号化処理した暗号化ウェブファイルに全般の認証を行う認証子を含むデータをヘッダーとして付加した改竄防止ヘッダー付き暗号化ウェブファイルを格納する改竄防止機能付き暗号化公開ウェブサーバーコンピュータ、

(2) 前記改竄防止機能付き暗号化公開ウェブサーバーとファイアウオール等の不正アクセスを排除する手段を介して接続し、前記ウェブファイルを格納する原本ウェブサーバーコンピュータ、

(3) 公開ウェブサーバーコンピュータにおけるウェブサーバーが、ユーザーからアクセス要求を受けたとき、前記改竄防止ヘッダー付き暗号化ウェブファイルから該ヘッダーを分離し、該ヘッダー情報中の認証子で分離されたウェブファイルに対して改竄チェックをするリアルタイムチェック手段、

(4) 前記公開ウェブサーバーが、前記リアルタイムチェックにより前記ウェブファイルが改竄されていないと判断した場合、前記ヘッダーを除去したウェブファイルを復号化してユーザーに送信する手段、

(4) 前記改竄防止機能付き暗号化ウェブファイルの改竄を検出したとき、前記原本ウェブサーバーコンピュータで、記憶装置に格納されている対応するウェブファイルを暗号化処理し、改竄防止ヘッダーを附加して作成した改竄防止ヘッダー付き暗号化ウェブファイルにより、前記改竄防止機能付き暗号化公開ウェブサ

一バーコンピュータの記憶装置へ送信して改竄されたウェブファイルを更新・復旧処理する手段、
を含むことを特徴とするホームページ改竄防止システム。

【請求項4】

- (1) html、HTML、TEXT、GIF、JPEGなどの拡張子を有する画像、音声など静的なウェブファイル、及びユーザのブラウザ側で実行できるウェブファイルを暗号化処理した暗号化ウェブファイルを格納する公開ウェブサーバーコンピュータ、
- (2) 前記公開ウェブサーバーコンピュータにおいて、ユーザのブラウザからのCGIプログラムを実行させるURL形のリクエスト情報（IPアドレス、コメント、パラメータなど）を受け取り、必要な通常処理をする前に、CGIプログラムを直接に実行させずに、前記リクエスト情報データをCGI Gatewayモジュールへ渡す手段、
- (3) 前記CGI Gatewayモジュールが、原本ウェブサーバーが受信できるように、前記リクエスト情報データを自動に変換し、該変換されたリクエスト情報データを原本ウェブサーバー送信する手段、
- (4) 前記CGI Gatewayモジュールから送られた前記変換されたリクエスト情報データにより、前記原本ウェブサーバーコンピュータの記憶装置に格納されCGIプログラムを実行させる原本ウェブサーバー、
- (5) 前記原本ウェブサーバーがhttpヘッダとCGIプログラムの出力結果を公開ウェブサーバーコンピュータに置ける前記CGI Gatewayモジュールへ送信する手段、
- (6) 前記CGI Gatewayモジュールが前期CGIの出力結果を前記公開ウェブサーバー経由、あるいは直接にユーザのブラウザへ送信する手段。

を含むことを特徴とするホームページ改竄防止システム。

【請求項5】

前記暗号化および復号化処理がカオス暗号法により行われることを特徴とする請求項1乃至4記載のホームページ改竄防止システム。

【請求項6】

前記リアルタイムチェック手段がカオス理論を用いたメッセージ認証技術を用

いた手段により行われることを特徴とする請求項2至4記載のホームページ改竄防止システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、インターネットにおけるウェブサーバーコンピュータシステムに関する。さらに詳細には、ホームページ改竄防止コンピュータシステムに関する。

【0002】

【従来の技術】

インターネットは、基本ルール（TCP/IPプロトコル）に従えば、ネットワーク上の任意の場所にアプリケーション環境を構築できる。このシステムの柔軟性がインターネットの利点であるが、逆にそれがハッカー（システム不正侵入者）の標的になりやすいというセキュリティ上の弱点がある。インターネットは、基本ルール（TCP/IPプロトコル）にしたがえば、ネットワーク上の任意の場所にアプリケーション環境を構築できる。このシステムの柔軟性がインターネットの利点であるが、逆にそれがシステム不正侵入者の標的になりやすいというセキュリティ上の弱点も持っている。

【0003】

ウェブサーバーはhtml、HTML、TEXT、GIF、JPEG、画像、音声など静止的なウェブファイル及びユーザブラウザ側で実行できる.exeのような実行ファイルなどのウェブファイルをハードディスク等に保存している。ウェブサーバーは、ブラウザからの請求が来たら、関連するファイルを送り出すという基本仕組みを持っている。システム不正侵入者が何らかの方法でウェブサーバーに侵入すれば、簡単にhtmlなどウェブサーバー内のファイルを改竄できる。

【0004】

このような攻撃に対しては、ファイアウォール内にウェブサーバーを設置して、認証されたユーザーのみがウェブサーバーを参照できるようにすることで、不正侵入者の侵入を防ぎ、ウイルスに汚染されることを防止することができると考えられている。

【0005】

しかし、インターネットのウェブサーバーは多くの人に見てもらうために、ファイアウォール外に置く必要があり、このことがシステム不正侵入者の攻撃に対して弱い原因になっている。さらに、インターネットで連結されたウェブブラウザとウェブサーバーの間のhttp通信プロトコルは、TELNET・FTPのようなアプリケーションと異なって、リクエストに対して直接回答をしてセッションを切断するというようなアーキテクチャで作られているので、身元認証と追跡が非常に困難である。

【0006】

システム不正侵入者の攻撃に対抗する改竄防止技術のひとつとして、ホームページを常に監視するホームページ監視システムがある。基本的な原理は、ウェブサーバーに置かれているウェブファイルを、常時、ホームページをチェック（監視）している。図1に一例を示す。この例では、

P₁ → P₂ → P₃ → → P_i → → P_s → → P_n → P₁ → P₂ → の順にサイクリックにページデータをチェックしている。改竄が認められた時点で、そのウェブファイルの公開中止あるいは元データで置き換えるなどの対策を行う。

【0007】

【発明が解決しようとする課題】

上記のホームページ改竄防止方法の問題点として、ウェブサイドの規模（ウェブファイル数）およびコンピュータのパワー性能にもよるが、一般的に、すべてのウェブファイルを全部チェックするため、数分から数十分かかる。この間に、改竄されたファイルに対してユーザからのアクセス要求があると、改竄されたままのファイルが送り出されてしまう。

【0008】

すなわち、ページデータが増えた場合、全ウェブファイルを1回チェックするだけ10分とか数十分も掛かってしまうことであるかもしれない。たとえば、図1の例で1サイクルのチェック時間が10分とした場合、P_sのデータをチェックしているときにP_i (i < s) のデータが改竄されたとすると、P_sからP_iに

次のチェックが回ってくるまでの数分間は、改竄データがそのままインターネットユーザーに開放された状態になっていることであるすなわち、この方法では100%の改竄防止は不可能である。

【0009】

この従来の改竄防止方法では、監視サーバーは常時稼動しなければならないので、システムに大きな負荷がかかる。この結果、ウェブサーバーのレスポンスが低下して、CPUのパワーの不足といったような不測の事態を招く恐れがある。以上のような事情から、このようなチェックシステムは、サーバー内の全ファイルのチェックに時間がかかる大型システムに向いていないといえる。

【0010】

本発明が解決しようとする課題は、

- ①改竄した行為があったとしても、改竄されたウェブファイルおよびC G I プログラムの実行結果を外（アクセス者）へ送り出すことが絶対にあり得ないこと、
- ②システム不正侵入者がウェブサーバーに侵入したとしても、意味がある改竄ができること、
- ③改竄防止システムは停止しないまま、ウェブサイドのホームページの日常更新を行うこと、
- ④異なるOS間に移植しやすいため、アプリケーション層で改竄防止システムを構築すること、
- ⑤すでにウェブサイドを使っているユーザに対して、既存のウェブサイドを修正しなくとも容易に導入し易いこと

などの事柄を実現する、経済的、技術的に優れ、容易に導入できるホームページ改竄防止システムを開発することである。

【0011】

【課題を解決するための手段】

上記の課題を解決するために、請求項1に記載された本発明は、（1）ウェブファイルを暗号化処理した暗号化ウェブファイルを格納する公開ウェブサーバーコンピュータ、

（2）前記公開ウェブサーバーコンピュータとファイアウォール等の不正アクセ

スを排除する手段を介して接続し、前記ウェブファイルを格納する原本ウェブサーバーコンピュータ、（3）公開ウェブサーバーコンピュータにおけるウェブサーバーが、ユーザーからアクセス要求を受けたときに前記暗号化されたウェブファイルを改竄チェックを行い、改竄されていないと判断した場合、前記暗号化されたウェブファイルを復号化してユーザーに送信する手段、（4）前記公開ウェブサーバーが前記暗号化されたウェブファイルの改竄を検出したとき、前記原本ウェブサーバーコンピュータで、記憶装置に格納されている対応するウェブファイルを暗号化処理して作成した暗号化ウェブファイルにより、前記公開ウェブサーバー・コンピュータの記憶装置へ送信して改竄されたウェブファイルを更新・復旧処理する手段、を含むホームページ改竄防止システムである。

【0012】

請求項2に記載された発明は、（1）ウェブファイルに全般の認証を行う認証子を含むデータをヘッダーとして付加した改竄防止ヘッダー付きウェブファイルを格納する改竄防止機能付き公開ウェブサーバーコンピュータ、（2）前記改竄防止機能付き公開ウェブサーバーコンピュータと不正アクセスを排除するファイアウオール等の不正アクセスを排除する手段を介して接続し、前記ウェブファイルを格納する原本ウェブサーバーコンピュータ、（3）前記公開ウェブサーバーコンピュータにおけるウェブサーバーが、ユーザーからアクセス要求を受けたとき、前記改竄防止ヘッダー付きウェブファイルから該ヘッダーを分離し、該ヘッダー情報中の認証子で分離されたウェブファイルに対して改竄チェックをするリアルタイムチェック手段、（4）前記公開ウェブサーバーが、前記リアルタイムチェックにより前記ウェブファイルが改竄されていないと判断した場合、前記ヘッダーを除去したウェブファイルをユーザーに送信する手段、（5）前記改竄防止ヘッダー付きウェブファイルの改竄を検出したとき、前記原本ウェブサーバーコンピュータで、記憶装置に格納されている対応するウェブファイルに改竄防止ヘッダーを附加して作成した改竄防止ヘッダー付きウェブファイルにより、前記改竄防止機能付き公開ウェブサーバーコンピュータの記憶装置へ送信して改竄されたウェブファイルを更新・復旧処理する手段、を含むホームページ改竄防止システムである。

【0013】

請求項3に記載された発明は、（1）ウェブファイルを暗号化処理した暗号化ウェブファイルに全般の認証を行う認証子を含むデータをヘッダーとして付加した改竄防止ヘッダー付き暗号化ウェブファイルを格納する改竄防止機能付き暗号化公開ウェブサーバーコンピュータ、（2）前記改竄防止機能付き暗号化公開ウェブサーバーとファイアウォール等の不正アクセスを排除する手段を介して接続し、前記ウェブファイルを格納する原本ウェブサーバーコンピュータ、（3）公開ウェブサーバーコンピュータにおけるウェブサーバーが、ユーザーからアクセス要求を受けたとき、前記改竄防止ヘッダー付き暗号化ウェブファイルから該ヘッダーを分離し、該ヘッダー情報中の認証子で分離されたウェブファイルに対して改竄チェックをするリアルタイムチェック手段、（4）前記公開ウェブサーバーが、前記リアルタイムチェックにより前記ウェブファイルが改竄されていないと判断した場合、前記ヘッダーを除去したウェブファイルを復号化してユーザーに送信する手段、（4）前記改竄防止機能付き暗号化ウェブファイルの改竄を検出したとき、前記原本ウェブサーバーコンピュータで、記憶装置に格納されている対応するウェブファイルを暗号化処理し、改竄防止ヘッダーを附加して作成した改竄防止ヘッダー付き暗号化ウェブファイルにより、前記改竄防止機能付き暗号化公開ウェブサーバーコンピュータの記憶装置へ送信して改竄されたウェブファイルを更新・復旧処理する手段、を含むホームページ改竄防止システムである。

【0014】

請求項4に記載された発明は、（1）html、HTML、TEXT、GIF、JPEGなどの拡張子を有する画像、音声など静的なウェブファイル、及びユーザのブラウザ側で実行できる動的なウェブファイルを暗号化処理した暗号化ウェブファイルを格納する公開ウェブサーバーコンピュータ、（2）前記公開ウェブサーバーコンピュータにおいて、ユーザのブラウザからのCGIプログラムを実行させるURL形のリクエスト情報（IPアドレス、コメント、パラメータなど）を受け取り、必要な通常処理をする前に、CGIプログラムを直接に実行させずに、前記リクエスト情報データをCGI Gatewayモジュールへ渡す手段、（3）前記CGI Gatewayモジュール

が、原本ウェブサーバーが受信できるように、前記リクエスト情報データを自動に変換し、該変換されたリクエスト情報データを原本ウェブサーバー送信する手段、（4）前記CGI Gatewayモジュールから送られた前記変換されたリクエスト情報データにより、前記原本ウェブサーバーコンピュータの記憶装置に格納されCGIプログラムを実行させる原本ウェブサーバー、（5）前記原本ウェブサーバーがhttpヘッダとCGIプログラムの出力結果を公開ウェブサーバーコンピュータに置ける前記CGI Gatewayモジュールへ送信する手段、（6）前記CGI Gatewayモジュールが前期CGIの出力結果を前記公開ウェブサーバー経由、あるいは直接にユーザのブラウザへ送信する手段を含むホームページ改竄防止システムである。

【0015】

請求項5に記載された発明は、前記暗号化および復号化処理がカオス暗号法により行われることを特徴とする請求項1乃至4記載のホームページ改竄防止システムである。

【0016】

請求項6に記載された発明は、前記リアルタイムチェック手段がカオス理論を用いたメッセージ認証技術を用いた手段により行われることを特徴とする請求項2至4記載のホームページ改竄防止システムである。

【0017】

暗号法がカオス理論を用いた暗号法であり、前記認証がカオス理論を用いたメッセージ認証技術を用いた場合は特に優れたホームページ改竄防止システムとなる。

【0018】

図2は本発明のシステムの全体的な概念を示す図である。本発明では、ウェブファイル全体に対する認証を行う。認証チェックで改竄を検知したとき、ウェブファイル全体を送信せず、また、認証チェックで改竄を検知した場合には、システム管理者に知らせる手段を備えておけば、改竄に対する対処が速やかに行える。当然、ログ（履歴）も記録することが好ましい。

【0019】

ウェブファイルの暗号化を行う本発明では、改竄データを意味のあるデータとして（書き換えた内容のままで）インターネットユーザー（ソフトウェアで表現すればブラウザ）に送信してしまうことはない。本発明ではウェブファイルを暗号化して保持し、ページアクセス要求を受けた時点で復号化し、ユーザーに送信する。この方法を用いれば、システム侵入者がページデータを書き換えたとしても、その内容が直接ユーザーに送信されることはない。なぜなら、ページデータは復号化してユーザーに送信されるので、復号化により意味不明の内容に変化するからである。侵入者が暗号化された形でページデータを書き換えない限り、その書き換えたデータが意味ある内容として送信されることはない。

【0020】

インターネットユーザーに開放するのは改竄防止機能付き公開ウェブサーバー 컴퓨터に納められたウェブファイルであり、原本ウェブサーバーコンピュータに納められたウェブファイルは維持管理用およびバックアップ用として保管、管理する。すなわち、ホームページの更新や追加などがある場合には、まず原本ウェブサーバーコンピュータに納められたウェブファイルに対して更新・追加処理をし、動作確認を行う。その後、暗号化して公開ウェブサーバーコンピュータの改竄防止機能付きウェブファイルに移す。原本ウェブコンピュータ内のウェブファイルを直接インターネットユーザーに開放することはない。

【0021】

先に、改竄が「あり」と判断された場合には、改竄データを含むウェブファイルに対して、原本ウェブサーバーコンピュータと公開の改竄防止機能付きウェブサーバーコンピュータの形態をとっているので、改竄データを元のデータで差し替える方法が可能である。即ち、自動的に復旧することができる。

【0022】

認証(authentication)は、情報の正当性・完全性を確保するための技術である。暗号では情報の秘匿を目的とするが、認証では情報が変えられていないことの確認を目的とする。認証は、メッセージ認証、ユーザ認証、端末認証、時刻認証などに別れる。特に、ある情報を確かに生成したことを保証する方式をデジタル署名(digital signature)という。Chaos MAMとは、GCCカオス暗号法を用いた

新しいメッセージ認証法(Message Authentication Method)である。

【0023】

すなわち、原本ウェブサーバーコンピュータ内で元の平文Mに対して、ChaosMACのような認証技術を用いて平文Mのメッセージ認証子MACを作成しておき、公開の改竄防止機能付きウェブサーバーコンピュータで作成するメッセージ認証子MAC'をMACと照合して異なる場合には「改竄あり」と判断し、公開の改竄防止機能付きサーバーから原本のウェブサーバーコンピュータにMを要求して、改竄されたM'をMで置き換える。

【0024】

これに対しては、メッセージ認証を行うことで防ぐことが出来る。図3に示すように、メッセージ認証は、送信側で送信メッセージと暗号鍵（秘密鍵）からメッセージ認証子MAC (Message Authentication Code) を作成し、メッセージとMACを送信する。受信側は受信したメッセージM'（改竄されている可能性があるからMとは限らない）と、保持している鍵（秘密鍵あるいは公開鍵）からメッセージ認証子MAC'を作成し、MACとMAC'をチェックし、等しければメッセージの正当性が証明され、等しくなければメッセージに何らかの改竄があったと判断できる。

【0025】

図4に、本発明の暗号化ウェブファイルの構造を示す。附加されたヘッダー情報には、当該ファイルの認証子MAC、サイズ、日付、属性、保存場所などの情報が書き込まれる。本発明のシステムでは、カオス暗号法、およびカオス理論によるカオス認証技術は、速度の面で最も優れたものであるが、他の暗号方法及びメッセージ認証技術でも原理的には可能である。

【0026】

なお、認証子MACで行う方法を実施例で説明する。また、改竄されたウェブファイルデータをリアルタイムに修復してユーザーに送信する方法も実施例で説明する。

【0027】

本発明のリアルタイムチェック機能を有するウェブサーバーの原理を説明する

。周知の通り、ウェブサーバーの主な仕事は、リクエストされたホームページなどのウェブファイルをクライアント、つまりウェブブラウザに送ることだ。殆どの場合、リクエストされたウェブファイルはハードディスクに保存されていて、サーバから簡単にアクセスできるようになっている。ウェブサーバーはリクエストに応じてファイルを探し出して、その内容をリクエストしてきたクライアントシステムにhttpプロトコルで転送する。

【0028】

通常のウェブサーバーの原理を図5に示す。即ち：

- 1) 環境変数を読み取りなどの初期化処理
- 2) ウェブブラウザからのURLのようなhttpのリクエストを受け取る
- 3) 必要の処理する後、リクエストされたファイルをハードディスクから読み込み
- 4) リクエストされたファイルのコンテンツをウェブブラウザへ返信する。

【0029】

本発明では、図6に示すように、リアルタイム・チェック・モジュール、上記のウェブファイルをハードディスクからコンピュータのメモリに読み込むOpenfileモジュールと送信モジュールの間に差し入れることで、ホームページ改竄防止システムのエンジンになった。

【0030】

このリアル・タイム・チェック・モジュールの原理は、図7に示している。リアルタイムチェックモジュールでは、まずリクエスト情報によって、ハードディスクに格納する暗号化され、まだ認証子など情報が構成した改竄防止ヘッダー付けファイルをメモリに読み込む；

【0031】

メッセージ認証技術によってファイルは改竄されたかどうかをチェックする；もし、改竄されていなければ、改竄防止ヘッダー部分を切り落とし、復号化を行った後、ウェブブラウザへ送信をする。

【0032】

改竄された場合は、原本ウェブサーバーコンピュータにある復旧サーバーへ復

旧要求を出す。復旧サーバーが、リクエスト情報に指定された原本ウェブサーバーコンピュータのディレクトリに格納した元ファイルに対して、暗号化を行い、さらにメッセージ認証技術で認証子MACを生成、当該ファイルのサイズ、日付、時刻、属性などの情報と一緒に改竄防止ヘッダーに組み入れて、当該ファイルにつける。このファイルを公開ウェブサーバコンピュータへ送り出す。これによって、公開ウェブサーバコンピュータのに格納する改竄されたファイルを修復・更新できる。さらに、この更新（或いは修復）したファイルをウェブブラウザへ送信する。

【0033】

本発明のシステムでは、改竄されたファイルが、送信する時点に、必ずメッセージ認証チェックによって検出されるので、原理的に改竄されたファイルがウェブクライアントへ送信されることはない。

【0034】

本発明のシステムは、いわゆる「リアルタイムチェック」技術を初めて実現できたものである。即ち、本発明の改竄防止ウェブサーバーは、リクエストされたファイルだけ、かつ、それを外に送信するまえにチェックを行うため、コンピュータのCPUに負荷をあまりかけることがない。

【0035】

ただし、実用的なリアルタイムチェック技術を実現するためには、高速の暗号と高速かつ強力な認証チェック技術が必要条件である。そこで、本発明では、高速な暗号として知られているGCCカオス暗号とカオスMAM認証技術によって、最高レベルのホームページ改竄防止システムを実現することが出来る。

【0036】

前記リアルタイムチェック技術はhtml、HTML、TEXT、GIF、JPEGなどの拡張子を持つ、画像、音声など静的なウェブファイル、及びユーザブラウザ側で実行できる実行ファイルに対して大変有効な方法である。

【0037】

しかし、CGIファイルに対しては他の方法を考えなければ成らない。CGI (Common Gateway Interface) とは、ウェブサーバーが実行できるプログラムである。

CGIスクリプト、あるいはCGIファイルとも読んでいる。CGIは言語に依存しないゲートウェイインターフェイス仕様であり、CやC++、Perl、さらにはJAVAなど実質的には一般に使用されているアプリケーション開発言語を使ってインプリメントできる。一般にその拡張子を.pi、.cgi、.exeのように定義されている。

【0038】

CGIプログラムは、ウェブサーバーの管理者、使用者達は特有な機能を追加するため開発するものである。使用方法はいろいろある。たとえば、別のコンピュータにあるDATABASEサーバーからDATAを取り出したり、編集（例えば、合計、統計処理、グラフィック作成など）して、結果を送り出すという使用方法があるし、より複雑な使う方法もある。CGIプログラムはOpenTextのような別の実行ファイルを実行して、得られた結果をブラウザへ送信するという方法もある。

【0039】

CGIプログラムの実行は、基本的にクライアントから、URLの形式のリクエストを応じて、ウェブサーバーの環境で実行して、その結果をクライアントであるブラウザへ返信することである。こうした処理の流れは図8に示している。

- ① CGIプログラムのための環境変数を設定する。HTTPのリクエストメソッドの名前をREQUEST_METHODという環境変数に設定して、クライアントから受け取ったデータをQUERY_STRINGという環境変数に設定する。
- ② リクエストされたCGIプログラムを実行する
- ③ CGIプログラムが終了するのを待ち、その出力結果をSTDOUTから読み込んで解析し、出力のContent-Typeを突き止める。
- ④ 必要のHTTPヘッダを生成する。
- ⑤ リクエストを出したクライアントに、ヘッダとCGIプログラムの出力をブラウザへ送信する。

【0040】

本発明で提案したホームページ改竄防止システムは、公開ウェブサーバーコンピュータと原本ウェブサーバーコンピュータから構成される。原本ウェブサーバーコンピュータには通常のHTML、TEXT、GIF、JPEG、などの非実行ファイルおよびCGIというファイルが収納されている。原本ウェブサーバーコンピュータは通

常のウェブサイトとして稼動できるものであるが、本発明のシステムでは元ファイルの保存場所として利用している。

【0041】

公開ウェブサーバーコンピュータに置く非実行ファイルは暗号化され、認証子が付けられる。また、ウェブサーバーでのリアルタイムチェック技術を使うことで、ウェブサイトの改竄を防止することができる。ただし、CGIプログラムの場合、事情は変わる。CGIプログラムの実行は原本ウェブサーバーコンピュータのOS、IPアドレス、ディレクトリ構造などの実行環境に依存している。原本ウェブサーバーコンピュータから公開ウェブサーバーコンピュータに移動するとIPアドレス、ディレクトリ、などの実行環境は変わったのでうまく実行できないことが多い。従って、本発明では、CGIプログラムの問題をクリアできる新たな提案を以下のように示す：

【0042】

つまり、原本ウェブサイトには通常のウェブサーバー（例えば：Apache、Netscapeなど）を使い、通常のhtml、GIF、およびCGIなどの元ファイルが存在する。公開ウェブサーバーコンピュータには、CGIファイルを直接に実行しないように改造されたウェブサーバーがある。さらに、CGI Gatewayモジュールを追加する。また、公開ウェブサーバーコンピュータにはCGIファイルは置かないことである。

【0043】

本発明のCGIの処理の流れを図9に示す。

- ① 公開ウェブサーバーコンピュータでは、CGIプログラムのための環境変数を設定する。HTTPのリクエストメソッドの名前をREQUEST_METHODという環境変数に設定して、クライアントから受け取ったデータをQUERY_STRINGという環境変数に設定する。
- ② ブラウザから貰ったCGIプログラムのリクエスト情報（IPアドレス、コメント、パラメータなど）をCGI Gatewayモジュールへ渡す。
- ③ CGI Gatewayモジュールでは、貰ったリクエスト情報を原本ウェブサイトが受け入れるように自動修正を行い、原本ウェブサーバーへCGIファイルを実行す

るリクエストを送信する

- ④ 原本ウェブサーバーでは、通常の通り、リクエストされたCGIプログラムを原本コンピュータ上で実行させる
- ⑤ httpヘッダとCGIプログラムの出力を公開ウェブサーバーコンピュータのCGI Gatewayモジュールへ返信する。
- ⑥ CGI GatewayモジュールがCGIの出力結果を公開ウェブサーバー経由、あるいは直接にブラウザへ送信する。

【0044】

【発明の実施の形態】

【0045】

本発明の実施例をGCCカオス暗号法とChaosMAMカオス認証技術を用いた場合の例で説明する。まず、簡単にカオス暗号法を説明する。カオス暗号法では公開鍵暗号方式と共通鍵暗号方式が使えるが、ここでは共通鍵暗号方式で説明する。本発明の場合も、ユーザーに鍵を渡す必要がないために、共通鍵暗号方式で十分できる。いま平文P、カオス暗号関数G、暗号文C、暗号鍵Kとしたとき、

$$C = G(K, P)$$

と暗号化できる。暗号文Cを復号化するには、カオス暗号逆関数G⁻¹と鍵Kを用いて、

$$P = G^{-1}(K, C)$$

と復号化できる。カオス暗号法においては平文Pの長さは自由である。鍵Kの長さは可変長で、8から2048ビットである。

【0046】

本発明は、①公開ウェブサーバーと②エンコーダー/デコーダーモジュールと③復旧サーバーと復旧クライアントと④警報システムなど部分から構成。そこで、①公開ウェブサーバーでは通常のウェブサーバーの全部機能（例えば、Apache）+デコーダー機能、まだ、②エンコーダー/デコーダーモジュールでは暗号化、認証子付け、ヘッダー情報を付けるなど機能を実行するエンコーダー部分及び、認証チェック、復号化、ヘッダー情報を切れるなど機能を実行するデコーダー部分、③復旧Serverの中にエンコーダーの機能が入っている。

【0047】

図9は、本発明のウェブファイルを暗号化するホームページ改竄防止システムを概念的に示したシステム構成図である。基本的に図2に示したシステム構成と同じであるが、改竄防止機能付き公開ウェブサーバーコンピュータのウェブファイルが暗号化されていることと、復旧クライアントを介して暗号化ウェブファイルの更新を行っていること、および暗号化にカオス暗号法が用いられていることである。

【0048】

ホームページのHTMLファイルを含むウェブファイルを原本ウェブサーバーコンピュータの上に持ち、復旧サーバーを通じて、ウェブファイルをGCC暗号化とMAM認証子< MAM MAC : Message Authentication Methodによる認証子の意味>）生成、及びファイルサイズ、日付、認証子などを含むヘッダー情報部分を追加するというエンコーダーを行い、ファイアウォールの外側に設置した復旧サーバーの管理下にある復旧クライアントに送信する。

【0049】

復旧クライアントは、受信したエンコーダーされたウェブファイルを復旧サーバーの指示した場所に置く。ネットワークユーザーからページ要求があった場合、改竄防止機能付きウェブサーバーは、(1)エンコーダーされたウェブファイルのヘッダー部分から、認証子などの情報を読み出し、(2)認証チェックを行い、認証にパスしたときには、ヘッダー部分を切り捨て、復号化を行う、というデコーダー操作をして、元に復帰されたウェブファイルをユーザーに送り出す。

【0050】

もし認証で改竄が認められた場合には、復旧クライアントを通じて、復旧サーバーへ改竄されたファイルを更新する請求を出し、復旧サーバーが請求に応じて、指定ファイルを通常のサーバーから取り出して、エンコードして、改竄防止機能付きサーバーへ送り出す、というような復旧を行う。また同時に、警報サーバーにその旨を伝え、警報サーバーから公衆回路を通じてシステム管理者に不正侵入者の存在を知らせる。

【0051】

本発明のシステムは以下のようなメリットがある：

① 公開ウェブサーバーコンピュータの中に、CGIファイルは存在しないから、たとえハッカーがこの公開ウェブサーバーに侵入したとしてもCGIファイルを改竄することができない。因みに、公開ウェブサーバーコンピュータと原本ウェブサーバーコンピュータの間に通常ファイアウォールを使っているので、原本ウェブサーバーコンピュータへハッカーが侵入することができない。ゆえに、原本ウェブサーバーコンピュータの中のCGIファイルはまず安全だと考えている。

【0052】

② CGIプログラムは、原本ウェブサーバーコンピュータで、従来の環境のまま実行することができる。ウェブサード構築者にとっては、ホームページを更新するとき従来の操作方法ままに行えるので、本発明のホームページ改竄防止システムの導入及び日常の更新、メンテナンスは非常にやり易い。

③ ウェブサイトのホームページをアクセスする公開ユーザとして、公開ウェブサイトにCGIを実行させるリクエストを出す時、従来のままで行うことができる。

【0053】

④ C言語、C++、Perl、Javaなどコンピュータ言語で作った、さまざまのCGIに対応することができる。特にFastCGIにも対応することが原理的に可能である。

【0054】

【発明の効果】

本発明のシステムによれば、以下のような効果を奏すことができる。本発明は、ウェブリアルタイム・チェック技術を実現するものである。高速性の点では、ブラウザからの請求が来た瞬間に、認証チェック、復号を瞬時に行い、チェックシステムがない場合と比べてレスポンス速度はほとんど変わらない。安全性の点では、改竄行為があったとしても改竄されたファイルを外（ブラウザ側）に送り出さない。

【0055】

大型システムに対しても、(1) トライフィックを増加しないのでウェブサーバーに負荷を増加させない。(2) ホームページシステムの規模（ファイル数）が

増大しても、チェック時間と復旧速度に影響を与えない。本発明のシステムはブラウザに影響は与えない。従来のブラウザソフトはそのまま使用でき、新しいクライアントソフトをダウンロードする必要はない。本発明のシステムは、ダイナミックな復旧機能を持っている。改竄を発見すると、自動的に高速で元のファイルを入れ替える。また、自動警報機能を設けることもでき、改竄を発見すると、自動的にシステム管理者の携帯電話、ポケベルに発信する。さらに、本発明のシステムは導入しやすく、既存のホームページ編集システムに影響を与えない。本発明のシステムでは、公開ウェブサーバーコンピュータ1台を導入、ポリシー管理システムを既存のウェブサイトにインストール、簡単なセッティングをすればOK（基本のウェブServer管理知識が必要）である。

【0056】

本発明のシステムによれば、ホームページ編集者は慣れているツールを用いることができるので、現在のホームページの環境を変えずに、ホームページをデザインしたり、作成したり、更新したりする事が出来る。そして、自動的に最新のホームページのウェブファイルを暗号化、認証子を附加してウェブサーバーに送ることができる。

【0057】

本発明のシステムでは、(1)改竄行為があったとしても、改竄されたウェブファイルを外部（アクセス者）へ送り出すことがないこと、(2)ハッカーがウェブサーバーに侵入したとして、ウェブファイルが暗号化されている場合は意味がある改竄が出来ないことなどが実現できる。

【0058】

さらに、本発明の改竄防止システムは、現在のウェブサイト（すでにホームページを開設しているサイト）においてもシステムの全面的な修正を行わなくても容易に導入ができるというメリットがある。しかも、特別な装置や技術を必要としないために経済的負担も少ない。一方、ユーザーが用いるブラウザに対しては何の負担もなく、現在使用しているものがそのまま使える（インターネットへ送り出すときは従来通りのページデータであるため）。すなわちインターネットユーザーへの負担は一切ない。

【0059】

特に、カオス暗号法とChaos MAM認証技術を用いることによって、以下のような優れた効果が得られる。安全性が高い（新しい暗号法であり、暗号法が解読される可能性がきわめて低い）、処理速度が速く、システムへの負担が少なく、リアルタイムの処理が可能（暗号化、復号化、認証チェック速度が速い）、ブラウザソフトへの影響はゼロ。改竄防止の認証チェックはh t t pなどのウェブファイルがクライアントへ送信する要求が出された時点で行なうから、また、復号処理もその時点で行なうから処理速度が要求され、その暗号法としてもカオス暗号法は最適である。

【図面の簡単な説明】**【図1】**

従来技術における改竄チェック方法を説明する図である。

【図2】

本発明のシステムの概念を説明するためのシステム構成図である。

【図3】

本発明においてメッセージ認証を説明するための図である。

【図4】

本発明における暗号化ファイルの構造の説明図である。

【図5】

通常ウェブサーバーの構造の説明図である。

【図6】

本発明におけるリアルタイム・チェック・モジュールを追加したウェブサーバーの構造の説明図である。

【図7】

リアルタイム・チェック・モジュールの原理の説明図である。

【図8】

CGIプログラムの実行原理の説明図である。

【図9】

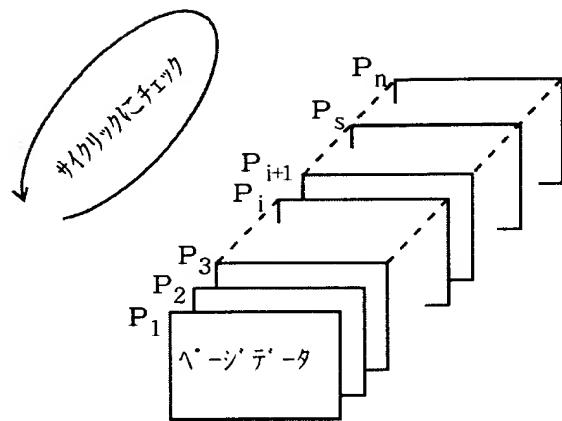
CGIファイルの改竄防止の原理の説明図である。

【図10】

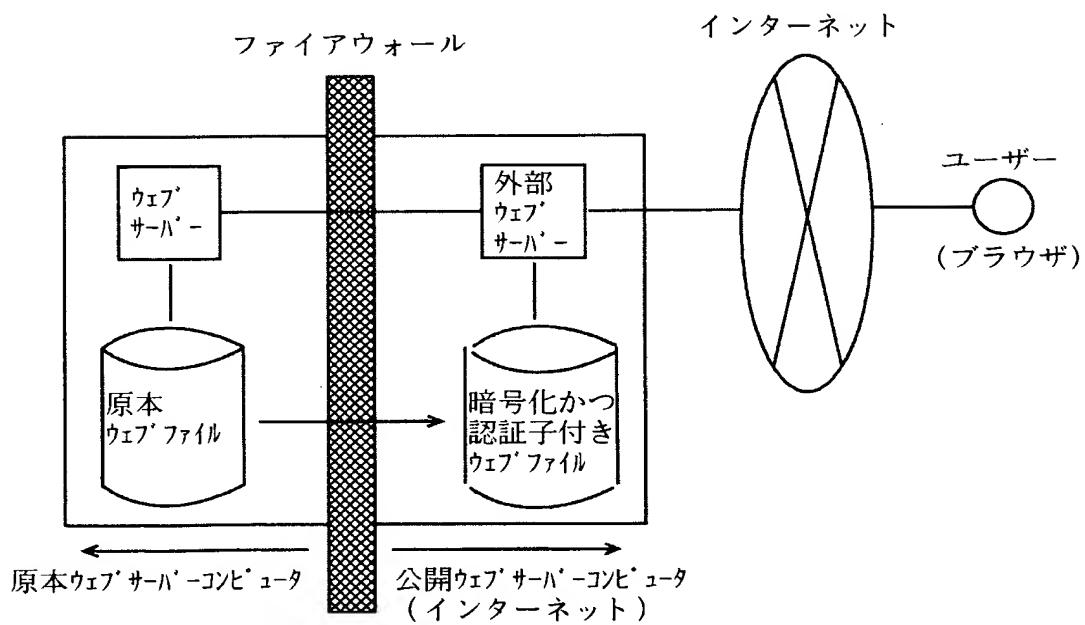
本発明の実施例におけるカオス暗号法を用いたホームページ改竄防止システムのシステム構成図である。

【書類名】 図面

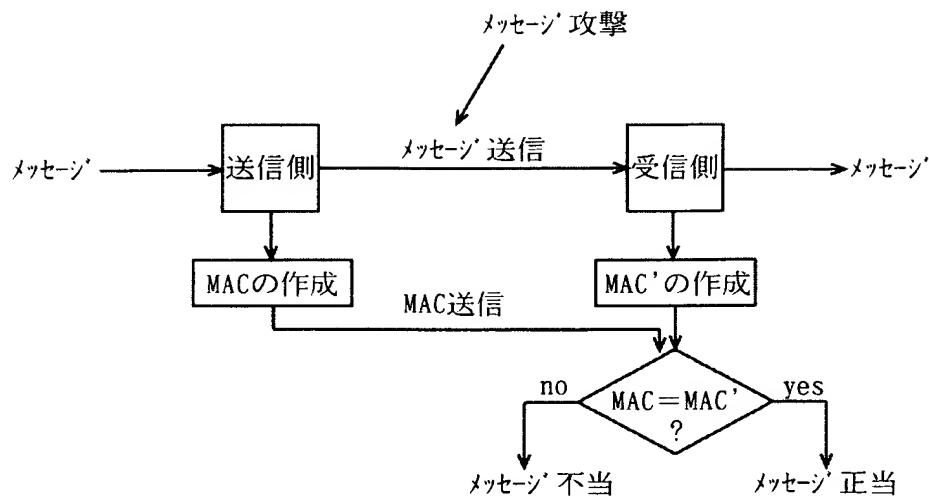
【図 1】



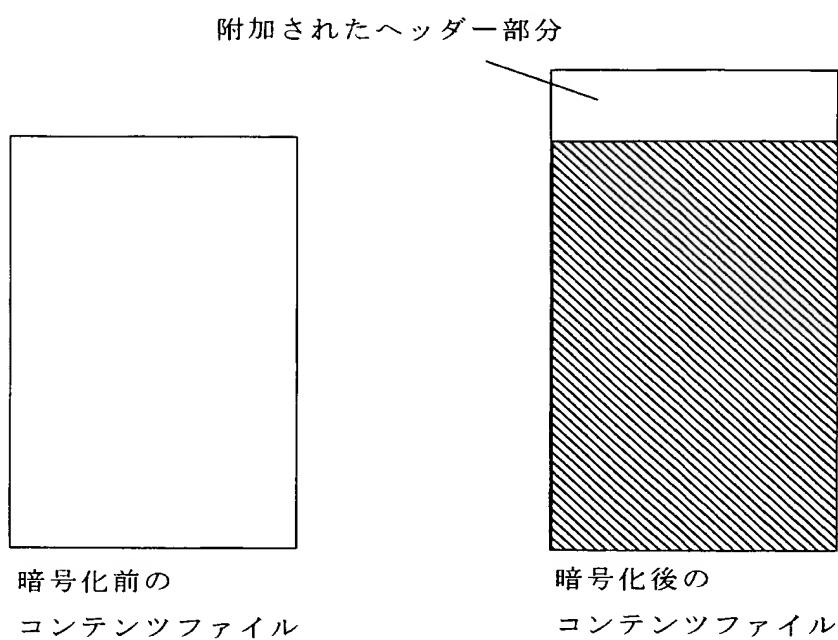
【図 2】



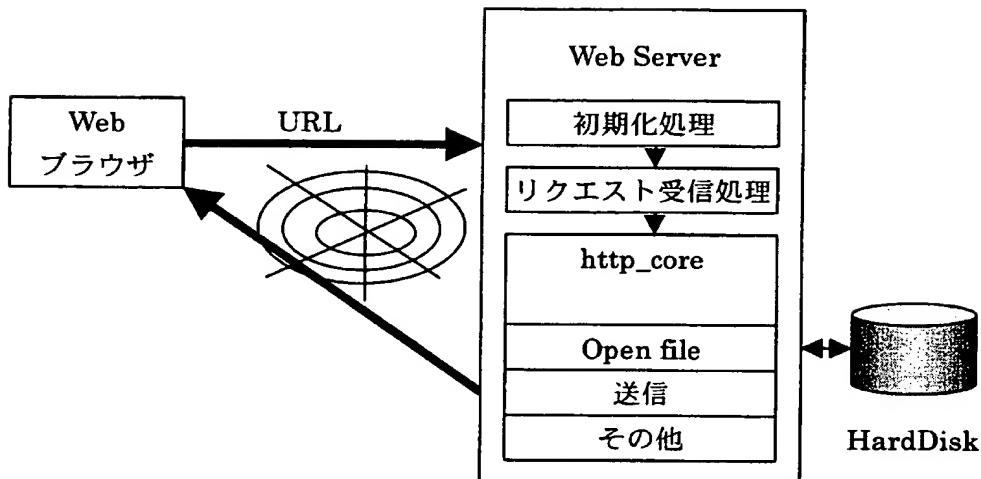
【図 3】



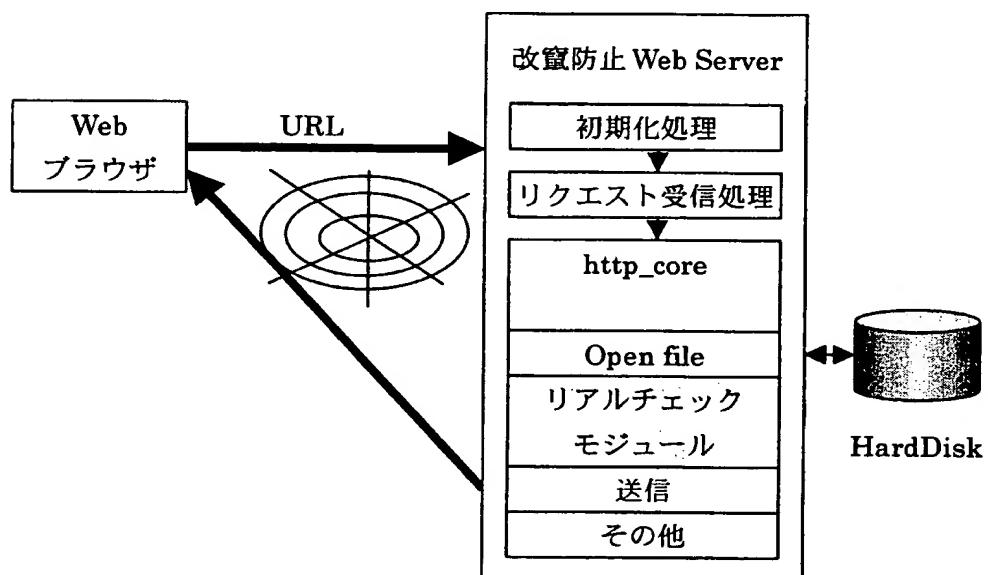
【図 4】



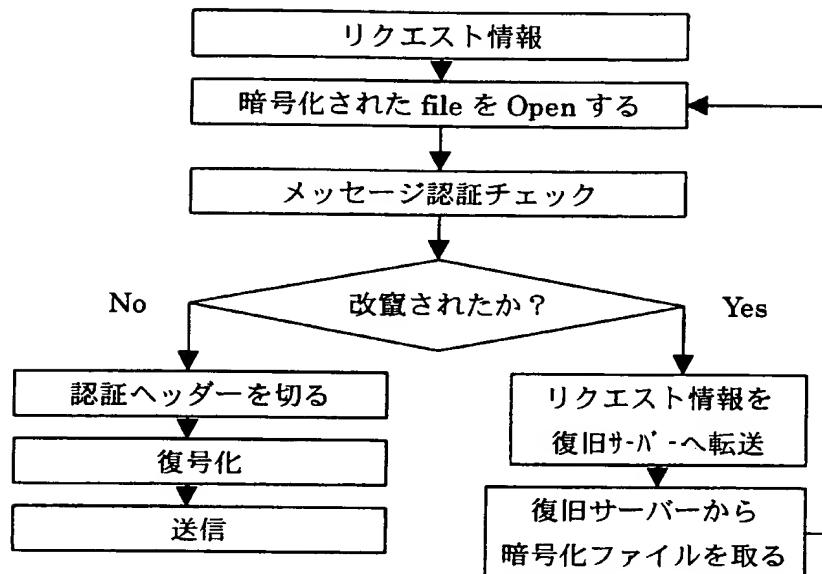
【図 5】



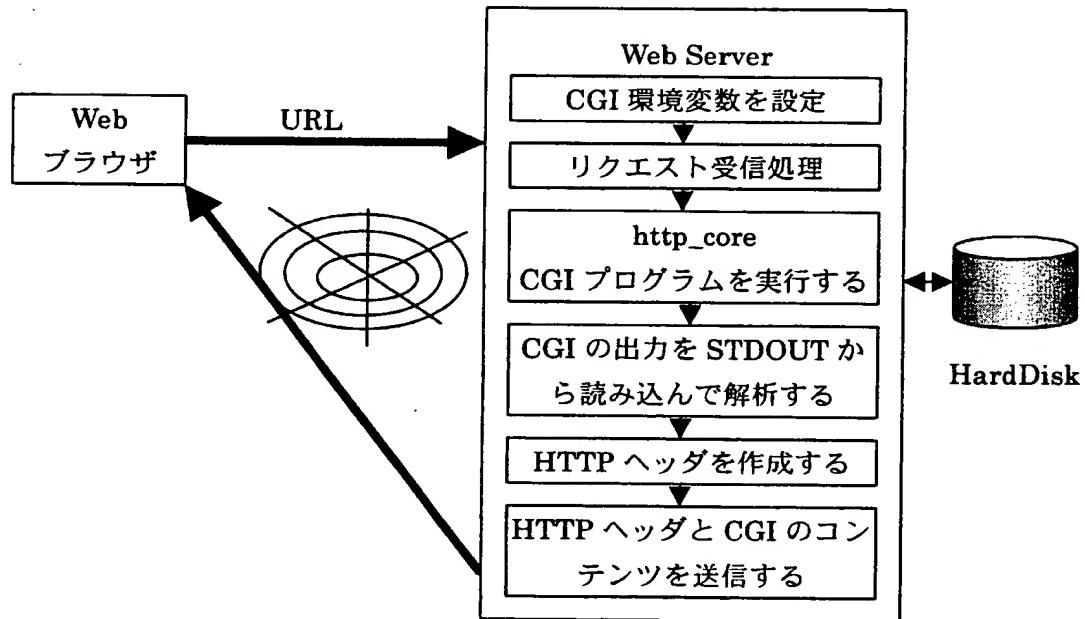
【図 6】



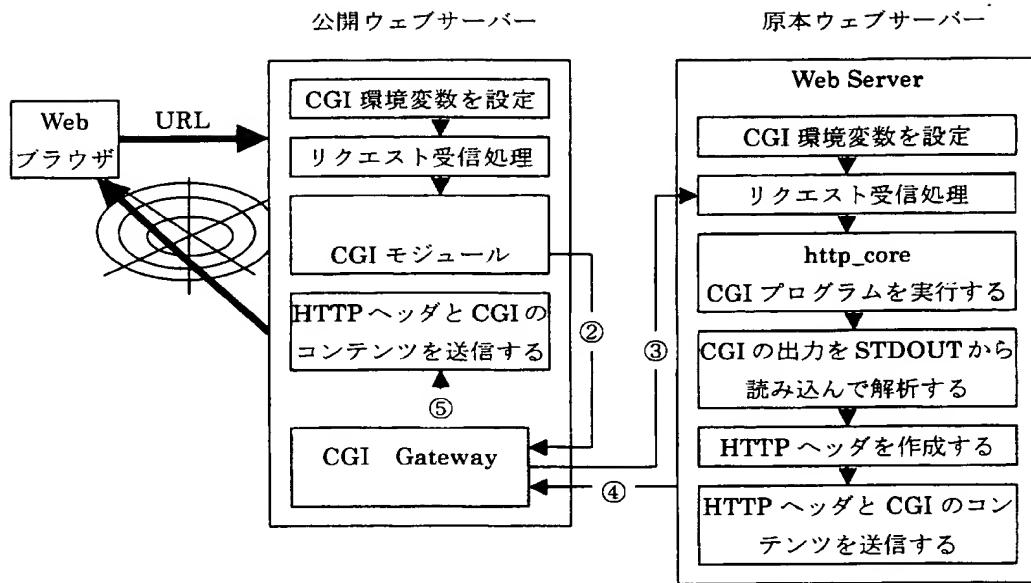
【図 7】



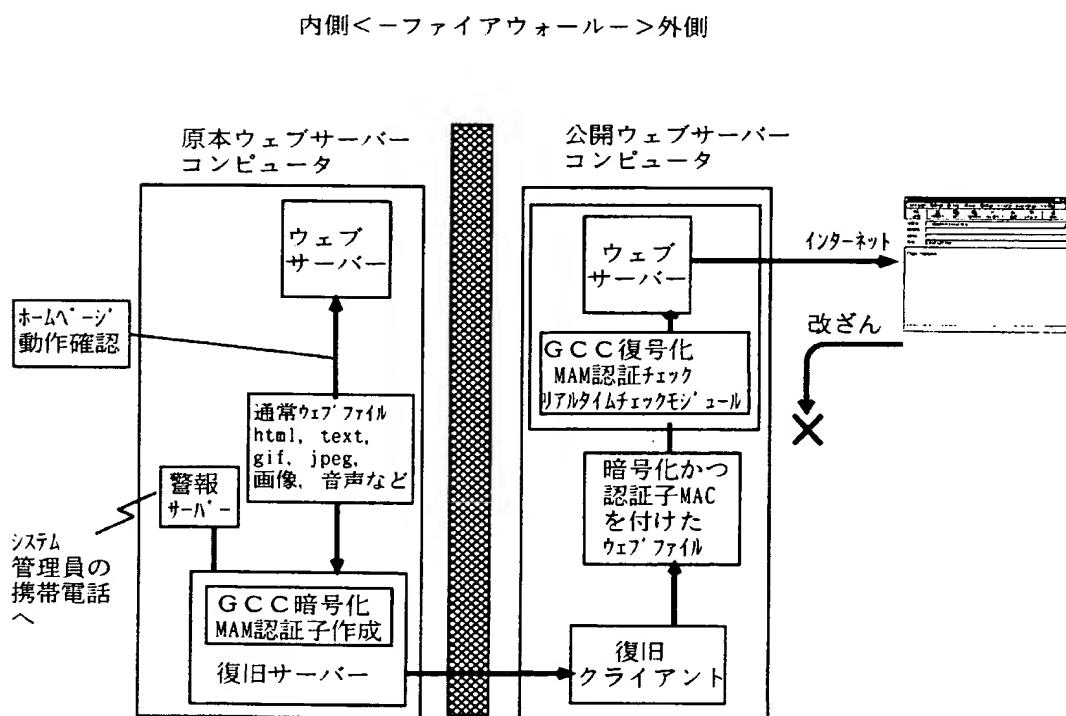
【図 8】



【図 9】



【図 10】



【書類名】 要約書

【要約】

【課題】 改竄した行為があったとしても、改竄されたウェブファイルおよびCGIプログラムの実行結果を外（アクセス者）へ送り出すことが絶対にあり得ないこと、システム不正侵入者がウェブサーバーに侵入したとしても、意味がある改竄ができないこと、改竄防止システムは停止しないまま、ウェブサイドのホームページの日常更新を行うこと、異なるOS間に移植しやすいため、アプリケーション層で改竄防止システムを構築すること、すでにウェブサイドを使っているユーザに対して、既存のウェブサイドを修正しなくとも容易に導入し易いことなど実現するホームページ改竄防止システム。

【解決手段】 ウェブファイルを暗号化処理した暗号化ウェブファイルを格納する公開ウェブサーバーコンピュータ、前記公開ウェブサーバーコンピュータとファイアウォール等の不正アクセスを排除する手段を介して接続し、前記ウェブファイルを格納する原本ウェブサーバーコンピュータ、公開ウェブサーバーコンピュータにおけるウェブサーバーが、ユーザーからアクセス要求を受けたときに前記暗号化されたウェブファイルを改竄チェックを行い、改竄されていないと判断した場合、前記暗号化されたウェブファイルを復号化してユーザーに送信する手段、前記公開ウェブサーバーが前記暗号化されたウェブファイルの改竄を検出したとき、前記原本ウェブサーバーコンピュータで、記憶装置に格納されている対応するウェブファイルを暗号化処理して作成した暗号化ウェブファイルにより、前記公開ウェブサーバー・コンピュータの記憶装置へ送信して改竄されたウェブファイルを更新・復旧処理する手段、を含むホームページ改竄防止システム。

【選択図】 図2

【書類名】 手続補正書

【整理番号】 K001-01

【提出日】 平成13年 6月13日

【あて先】 特許庁長官殿

【事件の表示】

【出願番号】 特願2001-161754

【補正をする者】

【識別番号】 593221598

【氏名又は名称】 高 振宇

【国籍】 中華人民共和国

【代理人】

【識別番号】 100093517

【弁理士】

【氏名又は名称】 豊田 正雄

【発送番号】 053719

【手続補正 1】

【補正対象書類名】 特許願

【補正対象項目名】 提出物件の目録

【補正方法】 追加

【補正の内容】

【提出物件の目録】

【包括委任状番号】 0015598

【プルーフの要否】 要

特願 2001-161754

出願人履歴情報

識別番号 [593221598]

1. 変更年月日 2000年10月10日

[変更理由] 住所変更

住 所 埼玉県川口市西青木1-23-28 ロイアルコーポ602室
氏 名 高 振宇